

REMARKS

I. Introduction

In response to the Office Action dated October 26, 2004, no claims have been cancelled, amended or added. Claims 1-57 remain in the application. Re-examination and re-consideration of the application is requested.

II. Prior Art Rejections

A. The Office Action Rejections

In paragraphs (3)-(4) of the Office Action, claims 1-3, 20-22, and 39-41 were rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin "Resource Management System for Multimedia Devices" (IBM) in view of Isip, Jr. et al., U.S. Patent No. 6,427,143 (Isip). In paragraph (9) of the Office Action, claims 4-6, 23-25, and 42-44 were rejected under 35 U.S.C. §103(a) as being unpatentable over IBM in view of Isip as applied to claims 1, 20, and 39, and further in view of Bordonaro et al., U.S. Patent No. 5,307,485 (Bordonaro). In paragraph (14) of the Office Action, claims 7-11, 26-30, and 45-49 were rejected under 35 U.S.C. §103(a) as being unpatentable over IBM in view of Isip as applied to claims 1, 20, and 39, and further in view of an "Official Notice" (ON). However, in paragraphs (16) of the Office Action, claims 12-19, 31-38, and 50-57 were indicated as being allowable if rewritten in independent form to include the base claim and any intervening claims.

Applicants' attorney acknowledges the indication of allowable claims, but respectfully traverse these rejections.

B. Applicants' Independent Claims

Applicants' independent claims 1, 20 and 39 are directed to loading data into a data store connected to a computer. Independent claim 1 is representative and comprises the steps of:

identifying memory constraints;

identifying processing capabilities; and

determining a number of load and sort processes to be started in parallel based on the identified memory constraints and processing capabilities.

C. The IBM Reference

IBM describes a resource management system for multimedia devices.

D. The Isip Reference

Isip describes data to be loaded into a table of a database that is read from an input file by a LOAD utility including a CHECK subutility that performs constraint checking on each row prior to loading the row into the table. During initialization of the CHECK subutility, the DB2 catalog is read to identify any constraints that apply to the data to be loaded. If any check constraints apply, check constraint routines are built and compiled by the CHECK subutility. If any referential integrity constraints apply, the key values for the parent index of each referential integrity constraint are read and stored in an additional dataspace allocated by the CHECK subutility so that the foreign key values from each row to be loaded can be located by the CHECK subutility and compared against the parent index values prior to loading the data into the table. When the constraint checking for the loading of data is complete, the additional dataspace allocated by the CHECK subutility is released.

E. The Bordonaro Reference

Bordonaro describes a system and method for merging a plurality of sorted lists using multiple processors having access to a common memory in which N sorted lists which may exceed the capacity of the common memory are merged in a parallel environment. Sorted lists from a storage device are loaded into common memory and are divided into a number of tasks equal to the number of available processors. The records assigned to each task are separately sorted, and used to form a single sorted list. A multi-processing environment takes advantage of its organization during the creation of the tasks, as well as during the actual sorting of the tasks.

F. Applicants' Independent Claims Are Patentable Over The References

Applicants' attorney respectfully submits that Applicants' claimed invention is patentable over the references. Specifically, Applicants' attorney asserts that the references do not teach or suggest the limitations recited in Applicants' independent claims 1, 20 and 39.

Nonetheless, the Office Action states the following:

5. Regarding claims 1, 20, and 39, IBM Bulletin discloses a method of processing data comprising the steps of:
identifying memory constraints (page 4, 1st paragraph: number of concurrently executing tasks is bounded by memory);

identifying processing capabilities (page 4, 1st paragraph: number of concurrently executing tasks is bounded by processor capabilities); and determining a number of tasks to be started in parallel based on the identified memory constraints and processing capabilities (page 4, 1st paragraph: number of concurrently executing tasks is bounded by memory and processor capabilities).

IBM Bulletin did not disclose the tasks are the load and sort processes. However, the concept of the load and the sort processes are considered well known in the art and can also be found in Isip, in which the load and sort processes are performed, based on the constraint factors (fig. 2, col. 6, lines 35 – 41, col. 12, lines 38 – 51). Isip also discloses that reader subtasks can run concurrently with other tasks of the load utility, such as a sort subtasks (col. 8, lines 60 – 67). It would have been obvious for an ordinary skill in the art to recognize the load and sort task of Isip can also be related to the tasks as disclosed by IBM Bulletin which perform a particular function. Therefore, it would have been obvious for an ordinary skill in the art, at the time the invention was made, to combine Isip's teaching together with IBM Bulletin to implement the tasks with the load and sort processes so that data can be processed more efficiently within the resource constraint.

Applicants' attorney disagrees. The cited portions of these references do not teach or suggest the combined limitations of Applicants' claims.

For example, the cited portions are set forth below:

IBM Bulletin: Page 4, 1st paragraph

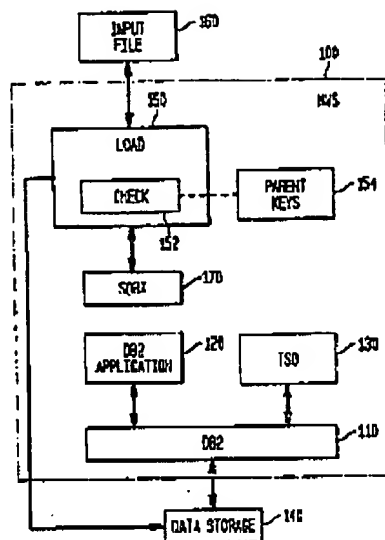
- For devices implemented with a (DSP), the number of concurrently executing DSP asks is bounded by the DSP memory and processor capabilities. Resource unit allocation for devices implemented on a DSP might be the following:

- FAX task - 10 resource units
- Audio task - 2 resource units
- MIDI task - 44 units

The problem of managing multiple logical devices implemented in hardware, such as a DSP or video transfer channel where hardware resources bound the number of device contexts of each device that can be concurrently supported, is addressed by varying the number of resource units each logical device can support. The multimedia resource manager provides an interface for the hardware device manager to specify the number of resource units a logical device can support.

When the number of resource units a logical device can support changes, the multimedia resource manager suspends or restores the device contexts for that logical device.

Isip: Fig. 2



Isip: col. 6, lines 35 – 41 (actually, lines 21-41)

For each record of the input file read by the LOAD utility 150, a check is made in step 3040 to determine if an end of file (EOF) condition has been reached. If an EOF condition is determined in step 3040, the LOAD utility according to an embodiment of the present invention proceeds to point C in FIG. 3B, described below. If an EOF condition is not reached in step 3040, however, then the LOAD utility 150 calls the CHECK subutility 152 in step 3050. The CHECK subutility 152 determines if there are any constraint violations in step 3060, illustrated in FIG. 3B and described in more detail with regard to FIGS. 4A, 4B and 5. If the CHECK subutility 152 determines that there is a constraint violation in step 3060, then in step 3070 the LOAD utility 150 discards the row and the row is not loaded into the table. If the CHECK subutility 152 determines in step 3060 that there is no constraint violation, then in step 3080 the LOAD utility 150 writes the row into the table (e.g., into the tablespace of DASD 140 allocated for storage of the table). In step 3090, the LOAD utility 150 extracts the index keys from the row to be loaded and passes the index keys to the SORT utility 170.

Isip: col. 12, lines 38 – 51 (actually, lines 38-63)

In contrast to conventional LOAD utilities such as illustrated in FIG. 6, the address space utilized according to an embodiment of the present invention includes the code for three separate SORT utilities 711, 712 and 713 instead of a single SORT utility as is used in conventional LOAD utilities. For example, as the processing speed (e.g., efficiency) for a SORT operation decreases non-linearly with an increase in the number of records to be sorted, providing a separate SORT utility (e.g., a separate SORT utility for each index in the order_entry table) can significantly increase the speed at which the SORT operation can be performed as well as allowing for the three SORT operations to be performed concurrently, thus further

improving the performance of the SORT operation. As three SORT utilities are provided, three sort buffers 721, 722 and 723 are also stored in the address space as shown in FIG. 7. The input file buffer 740, output file buffer 750 and index buffer 730 operate similarly to the corresponding buffers shown in FIG. 6. However, unlike the allocation and use of address space in prior art LOAD utilities, the LOAD utility according to an embodiment of the present invention does not need to provide a buffer to temporarily store extracted foreign keys for use in post-loading constraint enforcement as the pre-loading constraint enforcement using a CHECK subutility according to an embodiment of the present invention eliminates the need for temporarily storing foreign keys.

Isip: col. 8, lines 60 – 67 (actually, col. 8, line 60 – col. 9, line 7)

In step 4080, a reader subtask is attached to each parent index. The reader subtasks, as is known in the art, will start reading the indexspace for each parent index identified in step 4050 and will then store the parent key values into the dataspace 154 allocated for each parent index. The reader subtasks, as is known in the art, can run concurrently with other tasks of the LOAD utility 150, such as a SORT subtask. It should be understood that more than one dataspace 154 can be allocated by step 4080 so that there is a dataspace 154 for each parent index (corresponding to each referential integrity constraint defined for the table to be loaded and thus to be applied to the input file 160). Once a reader subtask has been attached to each parent index, the CHECK subutility 152 returns to the LOAD utility 150 in step 4090.

The descriptions set forth above do not teach or suggest all the limitations of Applicants' claims.

IBM merely describes the number of Fax, Audio and MIDI tasks that can be started on a single digital signal processor (DSP) based on the DSP memory and processor capabilities. Isip merely describes that three separate SORT utilities are used with a single LOAD utility. Nonetheless, the Office Action asserts that it would have been obvious to recognize that the LOAD and SORT utilities of Isip are related to the tasks disclosed by IBM, and that it would have been obvious to combine Isip's teaching together with IBM Bulletin.

Applicants' attorney disagrees. The Office Action fails to establish a prima facie case of obviousness, because there is no suggestion or motivation, in either reference themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the references or to combine the teachings of the references. See, e.g., M.P.E.P. §§ 706.02(j) and 2142. Instead, the suggestion comes from the Office Action itself, which constitutes impermissible hindsight.

Moreover, there must be a reasonable expectation of success, which is not present. IBM and ISIP are directed at two fields of endeavor, i.e., resource management for multimedia devices (IBM)

and loading data into tables (Isip). Because of these differences, there can be no expectation of success from the combination.

Finally, the prior art references, when combined, do not teach or suggest all the claim limitations. For example, Applicants' claims include the limitation of "determining a number of load and sort processes to be started in parallel based on the identified memory constraints and processing capabilities," but Isip starts only a single LOAD process (albeit three SORT processes), while IBM teaches neither load or sort processes at all.

As a result, neither of the IBM or Isip references teach or suggest all of the limitations of the independent claims. Consequently, it cannot be said that the combination of IBM and Isip teaches or suggests, or renders obvious, the Applicants' independent claims.

Bordonaro and ON fail to overcome the deficiencies of IBM and Isip. Recall that Bordonaro was cited only against dependent claims 4-6, 23-25 and 42-44, and ON was cited only against dependent claims 7-11, 26-30 and 45-49. Moreover, Bordonaro was cited only for teaching that the number of load processes does not exceed a number of partitions to be loaded, and that the load and sort processes directly dependent on memory constraints, and ON was cited only for teaching to efficiently utilize all processing capabilities required for the desired task. Neither of these teachings are relevant to the limitations of Applicants' independent claims.

Thus, Applicants' attorney submits that independent claims 1, 20 and 39 are allowable over the references.

G. Applicants' Dependent Claims Are Patentable Over The References

Dependent claims 2-11, 21-30 and 40-49 are submitted to be allowable over the references in the same manner, because they are dependent on independent claims 1, 20 and 39, respectively, and thus contain all the limitations of independent claims 1, 20 and 39. In addition, dependent claims 2-6, 21-25 and 40-45 recite additional novel elements not shown by the references.

With regard to claims 2, 21 and 40, which recite "determining a number of build processes based on the number of sort processes," the Office Action asserts that these limitations are described in Isip at Fig. 1 and col. 3, line 46 – col. 4, line 7. Applicants' attorney disagrees. At the indicated location, Isip merely describes a method for loading new records into a database and performing referential integrity checks, but says nothing about build processes.

With regard to claims 3, 22 and 41, which recite that "the number of sort processes does not exceed a number of indexes to be built," the Office Action asserts that these limitations are described in

Isip at Fig. 1 and col. 3, line 46 – col. 4, line 7. Applicants' attorney disagrees. At the indicated location, Isip merely describes a method for loading new records into a database and performing referential integrity checks, but says nothing about build processes.

With regard to claims 4, 23 and 42, which recite that "the number of load processes does not exceed a number of partitions to be loaded," the Office Action asserts that these limitations are described in Bordonaro in FIG. 3 (310 and 312), FIG. 2 (202), at col. 4, line 62 – col. 6, line 27, and col. 5, lines 58-60. Applicants' attorney disagrees. At the indicated locations, Bordonaro merely describes partitioning sorted lists into N tasks, and sorting N tasks on N processors.

With regard to claims 5, 24 and 43, which recite that "the total number of load and sort processes does not exceed processing capabilities," the Office Action asserts that these limitations are described in Bordonaro at col. 1, lines 57-68. Applicants' attorney disagrees. At the indicated location, Bordonaro merely describes sorting.

With regard to claims 6, 25 and 44, which recite that "the memory utilized by the load and sort processes does not exceed memory constraints," the Office Action asserts that these limitations are described in Bordonaro at col. 5, lines 54-62. Applicants' attorney disagrees. At the indicated location, Bordonaro merely describes the search bounds of the sorted lists read into common memory.

With regard to claims 7, 26 and 45, which recite that "the number of load processes and the number of sort processes each require different processing power," these claims stand or fall with independent claims 1, 20 and 39, respectively.

With regard to claims 8, 27 and 46, which recite that "the number of load processes and the number of sort processes each require similar processing power," these claims stand or fall with independent claims 1, 20 and 39, respectively.

With regard to claims 9, 28 and 47, which recite that "the number of load processes is not equal to the number of sort processes," these claims stand or fall with independent claims 1, 20 and 39, respectively.

With regard to claims 10, 29 and 48, which recite that "the number of load processes is equal to the number of sort processes," these claims stand or fall with independent claims 1, 20 and 39, respectively.

With regard to claims 11, 30 and 49, which recite that "the number of load processes is equal to the number of sort processes and which is equal to half of the processing capabilities," these claims stand or fall with independent claims 1, 20 and 39, respectively.

III. Conclusion

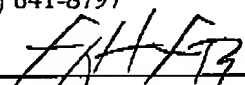
In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicants' undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicants

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: January 25, 2005

By: 
Name: George H. Gates
Reg. No.: 33,500

GHG/